

Peer Review

Review of: "Matching Frontier Code Agents with Lightweight Models via Multi-Model Consultation"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors ask whether frontier-level performance on SWE-bench Verified requires a frontier model, and answer that it does not. They run Claude Haiku 4.5 (Table 2: claude-haiku-4-5-20251001, temperature 0, 128K thinking budget, 250 max turns) over all 500 Verified instances under two policies: a plain single-agent **baseline**, and a **Polydev** policy in which the same agent may call a Model Context Protocol server that consults GPT 5.2 Codex and Gemini 3 Flash Preview (Table 3). Measured separately (Table 4), the baseline resolves $323/500 = 64.6\%$ and Polydev $333/500 = 66.6\%$. The paper then defines **Resolve@2** (Eq. 2-3, §3.2) as the union of the two resolved sets, $373/500 = 74.6\%$, and places that number alongside the December 15, 2025 leaderboard entry for Claude 4.5 Opus (medium), 74.4% (Table 1, rank 1). The cost analysis (Table 8, "Total Cost" row) puts both pipelines plus all consultation traffic at \$136.34, i.e. \$0.273 per instance, against an estimated \$0.72 per instance for Opus (Table 9, "Frontier comparison" row), which is the source of the abstract's "62% lower cost." A complementarity analysis (Table 5) reports Jaccard $J = 283/373 = 0.759$, discordance $90/373 = 0.241$, and McNemar $\chi^2 = (40-50)^2/90 = 1.11$, $p = 0.29$. Supporting material includes a consultation-impact taxonomy (Table 7, Table 11), per-repository results (Table 10, Appendix A Table 12), ablations on thinking budget, turn cap and consultation model (§6.4, Appendix C Table 13), and a released artifact repository.

I want to record at the outset that this paper does several things well: it labels Resolve@2 as an upper bound in the footnotes to Tables 1 and 4, it states plainly in §7.3 (limitation 3) that the Pass@2 ablation was never run, it declares the competing interest (V.S.G. is affiliated with Polydev AI), and it includes the consultation-impact labeling protocol in Appendix E with a Cohen's κ of 0.81. Those are not common

courtesies. My objections below are that the abstract, §1.3, and §10 do not honour the caveats that the body of the paper correctly states, and that a number of the tables do not reconcile with each other arithmetically.

2. *Whether the conclusion holds on the data*

2.1 The headline 74.6% is not a policy. It is the union of two runs, selected after the fact. Eq. 3 (§3.2) writes the metric explicitly as $\text{Rate}_{\text{Resolve@2}} = |B \cup P| / N$, and §4.4 gives the selection rule: "If Patch A (baseline) passes all tests, use Patch A; Else if Patch B (polydev) passes all tests, use Patch B." The tests in question are the SWE-bench FAIL_TO_PASS / PASS_TO_PASS suites, which are the very tests that define the resolution label. §3.2's disclaimer — "This is not an oracle with access to ground truth patches. Selection uses only test outcomes from the SWE-bench harness" — does not rescue the metric: with respect to the reported quantity, observing the harness verdict is observing the label. At deployment time no such verdict exists, so no system can be built that delivers 74.6%. The paper is internally inconsistent on this point: the footnotes to Tables 1 and 4 both call the row "Resolve@2 (oracle)", while §3.2 and §4.4 insist it is not an oracle. The deployable number in this paper is 66.6% (Table 4, row "Polydev (Multi-Model)", 333/500). Every top-level claim — abstract sentence 2, §1.3 contribution 1, §10 paragraph 1 — should be rewritten around 66.6%, with 74.6% presented only as a ceiling on what a perfect selector could reach.

2.2 The 74.4% Opus figure is quoted, not measured. Table 1 attributes it to the December 15, 2025 leaderboard snapshot for "Claude 4.5 Opus (medium)" — i.e. medium reasoning effort, a configuration the paper never matches. §7.3 (limitation 4) concedes this: "Our comparison uses leaderboard Opus results, which may use different configurations." The abstract and §10 drop both the "(medium)" qualifier and the fact that no Opus run was performed here. The comparison is therefore between a measured two-run union under one scaffold and a third-party single-run number under an unknown scaffold.

2.3 The intervals overlap almost completely, and the paper reports the overlap asymmetrically. Resolve@2 is 74.6% with CI [70.5%, 78.3%] (Table 4); Opus is 74.4% with CI [70.3%, 78.1%] (Table 1). These intervals share more than 98% of their span. In counts, 74.6% is 373/500 and 74.4% is 372/500: the "match" is a one-instance difference, comfortably inside the $\pm 4\%$ that §8.4 itself concedes. Yet §5.1 devotes a sentence to arguing that the Resolve@2 interval is "nearly separated" from Polydev's and "touching only at the boundary" — a comparison that is both statistically void and rhetorically one-sided. It is void because $B \cup P \supseteq P$ by construction (Eq. 2): Resolve@2 can never fall below Polydev on any

instance, so there is nothing to test. That sentence should be deleted, and the Resolve@2-vs-Opus overlap stated in its place.

2.4 The CI arithmetic is inconsistent between rows. I recomputed all Wilson intervals at $n = 500$ and checked each twice. Appendix D.1 specifies $z = 1.96$ and reports, for $\hat{p} = 0.746$, Lower = 0.705 and Upper = 0.783. At $z = 1.96$ the correct values are [70.61%, 78.22%], not [70.5%, 78.3%]. The published bounds are reproduced exactly by $z = 2.0$, which gives [70.52%, 78.29%]. The same holds for Table 1 rank 1: the published [70.3%, 78.1%] is the $z = 2.0$ interval for 372/500 ([70.31%, 78.10%]), whereas $z = 1.96$ gives [70.40%, 78.03%]. But Table 4 row 1 (323/500, [60.3%, 68.7%]) matches $z = 1.96$ exactly ([60.31%, 68.67%]) and not $z = 2.0$ ([60.22%, 68.75%]). Table 4 row 2 (333/500, [62.3%, 70.6%]) matches neither: $z = 1.96$ gives [62.35%, 70.59%] and $z = 2.0$ gives [62.26%, 70.67%], so the published interval takes its lower bound from one z and its upper bound from the other. This matters directly for §5.1, whose "touching only at the boundary (-70.5-70.6%)" claim is an artifact of mixing the two conventions.

2.5 McNemar is reported correctly and then read backwards. $\chi^2 = (40-50)^2/90 = 100/90 = 1.111$ with 1 df gives $p = 0.29$ (Appendix D.2, Eq. 12); I confirm the arithmetic. The paper reads this in Table 5, §3.3 and §5.2 as evidence for "balanced bidirectional complementarity." A non-significant result is absence of evidence for a difference, not evidence for balance; with $b = 40$ and $c = 50$ the test simply lacks power to distinguish the two policies. More consequentially, McNemar on $b = 40 / c = 50$ is exactly the right test for the question the paper does not ask: **is Polydev better than the baseline?** By the paper's own statistic, the +2.0 pp gain in Table 4 (323 $\rightarrow$ 333, reported there as "+3.1%" relative) is not statistically significant, $p = 0.29$. The continuity-corrected variant, $(|40-50|-1)^2/90 = 0.90$, gives $p = 0.34$, and the exact two-sided binomial on 90 discordant pairs gives $p \approx 0.34$. So the single-policy contribution of multi-model consultation — the paper's actual mechanism — is not shown to exist.

2.6 Is the comparison budget-matched? On the cost side, largely yes, and the authors deserve credit for it: the \$0.273/instance in Table 9 is the Hybrid column of Table 8, which does include both Haiku pipelines (\$127.34) plus the GPT 5.2 Codex consultations (\$7.70) and the Gemini 3 Flash consultations (\$1.30). I verified every cell: $1.2M \times \$1.75 + 0.4M \times \$14.00 = \$7.70$; $0.8M \times \$0.50 + 0.3M \times \$3.00 = \$1.30$; $\$57.76 + \$78.58 = \$136.34$; $\$136.34/500 = \$0.2727$; and $1 - 0.273/0.72 = 62.1\%$, so the abstract's "62% lower cost" is arithmetically sound *given its inputs*. On the accuracy side, it is not budget-matched at all: \$0.273 buys two runs whose union is 74.6% only under label-aware selection, while \$0.72 buys one deployable Opus run at 74.4%. The honest budget-matched statement available from Table 9 is that Polydev costs

\$0.236 per resolved instance against Opus's \$0.97, at 66.6% versus 74.4% resolution — roughly 76% cheaper per resolved instance and 7.8 pp less capable.

2.7 The cost table applies prompt caching to one column and not the other. This is the discrepancy I consider most serious in §5.5, and I checked it twice. Table 8's Baseline column lists 28.4M input and 9.8M output Haiku tokens; at the Table 2 list rates (\$1.00/\$5.00 per MTok) that is $\$28.40 + \$49.00 = \$77.40$, but the reported subtotal is \$57.76. Since output at list is \$49.00, the implied input charge is \$8.76 on 28.4M tokens, i.e., an effective **\$0.3085/MTok, a 69.2% discount** — not the "85%" reduction stated in the Prompt Caching note immediately below the table. The Polydev column lists 26.1M input and 8.7M output, and $26.1 \times \$1.00 + 8.7 \times \$5.00 = \$69.60$, against a reported subtotal of **\$69.58**: that column is priced at list, with essentially no caching discount, even though it runs the same agent with the same system prompt and strictly more context. Two columns of the same table are therefore computed by two different methods, which contradicts the claim that "Subtotals in Table 8 reflect actual billed amounts from API responses."

2.8 The 62% figure is not robust to symmetric accounting. The Opus \$0.72/instance denominator is, by the authors' own admission (§5.5 caveat and §7.3 limitation 6), "derived from leaderboard median token usage patterns and official pricing" — that is, at list price, with no caching credit, while the numerator's baseline half has been credited a 69% input discount. Using only in-paper quantities: the baseline token mix of Table 8 is 56,800 input and 19,600 output tokens per instance, which at the Opus list rates quoted in §4.1 (\$5.00/\$25.00) gives \$0.774/instance — reassuringly close to the paper's \$0.72 estimate, confirming it is a list-price figure. Granting Opus the same effective input rate the Baseline column implies (30.85% of list) gives \$0.578/instance, against which \$0.273 is **52.7% lower, not 62%**. I do not claim \$0.578 is the right number — Opus token counts are unmeasured here — only that the headline 62% rests on an accounting asymmetry that the paper does not flag and that a symmetric treatment moves by roughly ten percentage points.

2.9 The comparison the paper needs is Pass@2, and it does not exist. Because Resolve@2 costs two runs, its fair comparator is not one Opus run but two baseline runs under the same harness-based selection. §3.5 estimates $\text{Pass@2}_{\text{iid}} = 1 - 0.354^2 = 0.875$ (arithmetically correct) and rightly calls it an unattainable upper bound, then §7.3 (limitation 3) states that "without empirical Pass@2 data, we cannot definitively claim that multi-model diversity outperforms single-model variance." I agree with that sentence and note that it flatly contradicts §10, contribution 1, which asserts "a 10.0 percentage point improvement (64.6% → 74.6%) through multi-model consultation, demonstrating that model diversity

provides genuine complementary value *beyond stochastic retries*." Two things are wrong in that one sentence: the 10.0 pp comes from oracle selection over two runs, not from consultation (consultation's own effect is the non-significant +2.0 pp of Table 4), and "beyond stochastic retries" is precisely what §7.3 says was never measured.

2.10 The consultation-model ablation inverts once normalised. §6.4 reports "GPT 5.2 Codex only: +5 additional instances over baseline; Gemini 3 Flash Preview only: +2; Both models: +10 (synergistic effect)" and concludes that "using both consultation models provides nearly double the benefit of either alone." Appendix C Table 13 gives the sample sizes: the GPT-only and Gemini-only rows are $N = 100$, the "Both models" row is $N = 500$. Per 100 instances, that is +5.0 (GPT-only), +2.0 (Gemini-only), and +2.0 (both). On this normalisation, the two-model configuration is no better than Gemini alone and less than half as good as GPT alone — the opposite of the stated synergy. Either Table 13's N column is wrong, or the §6.4 conclusion must be withdrawn.

3. Reproducibility

The reproducibility material is above the median for this literature, and I note the genuinely useful items: the base model snapshot is dated (claude-haiku-4-5-20251001, Table 2 and §4.6), temperature is stated as 0 (Table 2), the evaluation window is given (November 28 – December 12, 2025, §4.5), the harness is pinned ("official SWE-bench evaluation harness version 1.1.0", commit a3e0c7d, §4.6), the hardware is specified (AWS EC2 r6i.2xlarge, §4.5), token logs are promised for recalculation (results/token_logs/, §4.6; results/billing/, §5.5), and Appendix C Table 13 maps each ablation to a config file. The following remain missing or self-contradictory.

Snapshot identifiers. Only the base model carries a dated snapshot. Table 3 lists the consultation models as gpt-5.2-codex and gemini-3-flash-preview — floating aliases, one of them explicitly a "Preview". Over a two-week evaluation window (§4.5), a preview alias can be re-pointed; without dated snapshots, the consultation arm is not reproducible even in principle.

Dataset split. §4.5 and Appendix B name "SWE-bench Verified (500 instances)" but give no dataset identifier or revision (e.g., a Hugging Face repo plus revision hash). Given §8.2's own contamination discussion, the exact split artifact should be pinned.

Code version. §4.6 lists "Commit Hash: [release-v1.0] (tagged at submission)" — a bracketed placeholder, not a hash. The agent scaffold itself (tool implementations of bash, read_file, edit_file, glob, grep in

§4.3) and the Polydev MCP server have no version at all, which is a notable gap given the competing-interest declaration.

Seeds and attempts. No seeds are reported. At temperature 0, this is defensible, but §8.1 concedes residual non-determinism without quantifying it — a single repeated run on a subset would bound it. The number of attempts per instance per policy is never stated; from Eq. 2, it is evidently 1, but it should be said. Nor is it reported how many instances hit the 250-turn cap or exhausted the 128K thinking budget, which the paper needs, since §3.4 and §6.1 both attribute failures to turn-budget exhaustion ("Time-sensitive problems where consultation latency hurt (8 instances)").

Per-policy configuration. Table 2 is titled "Base Model Configuration," and it is not said whether the 250-turn / 128K settings applied to both policies. Appendix C Table 13 deepens the confusion by mapping "Thinking 128K, N=500" to `results/baseline/` and "Turns 250, N=500" to `results/polydev/`, although §6.4 reports both as endpoints of *baseline* performance (64.6%). One of those two output paths is mislabelled.

Ablation percentages are not attainable at the stated N. §6.4 reports 58.2% → 61.8% → 64.6% for thinking budgets and 54.2% → 60.4% → 63.2% → 64.6% for turn caps. Table 13 puts every intermediate point at N = 100, where the resolution rate can only take integer percentages. 58.2%, 61.8%, 54.2%, 60.4%, and 63.2% are all unattainable on 100 instances. Either the N column is wrong or the percentages are.

Cost accounting method. The three-category caching model in §5.5 (cache reads at 0.1×, cache writes at 1.25×, uncached at 1×) is the right framework, but Table 8 reports only aggregate input and output token counts, so a reader cannot reproduce the subtotals — and, as shown in §2.7 above, the Baseline and Polydev columns do not use the same method. Table 8 should carry per-category token counts (cache_read, cache_creation, uncached input, output) for each column.

4. What specifically to fix

1. **Abstract, sentences 1-2, and §10, paragraph 1.** Rewrite so the headline number is the deployable single policy, 66.6% (Table 4, row "Polydev (Multi-Model)", 333/500). Resolve@2 = 74.6% must never appear without the words "union of two runs, selected by harness verdict, upper bound" attached, as the footnotes to Tables 1 and 4 already do. Replace "matching Claude 4.5 Opus" with an explicit statement that the two-run union of a lightweight model reaches the same ceiling as a single frontier run.

2. **§3.2, paragraph beginning "Important:", and the footnotes to Tables 1 and 4.** Resolve the terminology contradiction: the footnotes say "(oracle)" and §3.2 says "not an oracle". State the operative fact once: selection observes the FAIL_TO_PASS/PASS_TO_PASS verdict, which is the label the metric is computed from, so Resolve@2 is label-aware and cannot be instantiated at deployment time.
3. **§10, contribution 1.** Delete "a 10.0 percentage point improvement (64.6% → 74.6%) through multi-model consultation" and "beyond stochastic retries". Consultation's measured effect is 323 → 333 in Table 4, i.e. +2.0 pp absolute (+3.1% relative), which by the paper's own McNemar test (Appendix D.2, $p = 0.29$) is not significant. The phrase "beyond stochastic retries" is contradicted by §7.3, limitation 3.
4. **§5.1, final sentence.** Delete the claim that the Resolve@2 CI is "nearly separated" from Polydev's. Since $\text{Resolve@2} = B \cup P \supseteq P$ by Eq. 2, the comparison is degenerate. Replace it with the comparison that bears on the thesis: Table 4's Resolve@2 CI [70.5%, 78.3%] against Table 1's Opus CI [70.3%, 78.1%] overlap over more than 98% of their span, and 74.6% vs 74.4% is 373 vs 372 instances.
5. **Table 1 (rank 1 row), Table 4 (all three rows), Appendix D.1 Eq. 9–10.** Recompute every Wilson interval at a single, stated z . At the $z = 1.96$ that Appendix D.1 declares, $n = 500$ gives: 323 → [60.31%, 68.67%]; 333 → [62.35%, 70.59%]; 373 → [70.61%, 78.22%]; 372 (Opus, 74.4%) → [70.40%, 78.03%]. The published [70.5%, 78.3%] and [70.3%, 78.1%] are $z = 2.0$ values, and Table 4 row 2's [62.3%, 70.6%] mixes both conventions.
6. **Table 5 and §5.2, "Key Insight" paragraph; §3.3, McNemar paragraph.** Stop reading $p = 0.29$ as positive evidence of balance. Report instead that the test is underpowered on 90 discordant pairs, give the continuity-corrected value $((|40-50|-1)^2/90 = 0.90, p = 0.34)$ or the exact binomial ($p \approx 0.34$), and state explicitly what the test does establish: the Polydev-vs-baseline difference of +2.0 pp is not significant.
7. **Table 8, Anthropic block, Polydev column.** The subtotal \$69.58 equals list price ($26.1 \times \$1.00 + 8.7 \times \$5.00 = \$69.60$), whereas the Baseline subtotal \$57.76 implies an effective input rate of \$0.3085/MTok ($\$57.76 - 9.8 \times \$5.00 = \$8.76$, over 28.4M tokens). Apply one method to both columns and recompute the Hybrid column, the Total Cost row, and Table 9.
8. **§5.5, "Prompt Caching" paragraph.** The stated "~85%" effective input-cost reduction does not match the Baseline column, which implies 69.2%. Either correct the prose or correct the subtotal, and add per-category token counts (cache_read_input_tokens, cache_creation_input_tokens, uncached_input_tokens, output_tokens) to Table 8 so readers can verify.

9. **Table 9, "Claude 4.5 Opus" row, and the abstract's "62% lower cost".** The \$0.72/instance denominator is a list-price estimate while the numerator is partly cache-discounted. Either apply caching symmetrically or drop the percentage. For scale: at the baseline token mix of Table 8 (56,800 input / 19,600 output per instance) and the Opus list rates of \$4.1, an Opus instance is \$0.774; at the Baseline column's implied effective input rate it is \$0.578, against which \$0.273 is 52.7% lower rather than 62%.
10. **Table 9, all rows.** Add a budget-matched line. The defensible claim available from the paper's own numbers is Polydev at \$0.236 per resolved instance and 66.6% against Opus at \$0.97 per resolved instance and 74.4%. Also correct the Hybrid Cost/Resolved cell in Table 8: $\$136.34/373 = \$0.3655$, which rounds to \$0.366, not \$0.365.
11. **Appendix A, Table 12.** The table does not sum. Its twelve repository rows total $N = 442$, not the 500 shown in the Total row; the Baseline column sums to 278 against a stated 323; Polydev to 291 against 333; Hybrid to 332 against 373; Both to 257 against 283; Base Only to 21 against 40; Poly Only to 33 against 50. Separately, the Hybrid column is not the union of the other three in nine of twelve rows: django gives Both 155 + Base Only 8 + Poly Only 13 = 176 but reports Hybrid 188, and the same off-by-one to off-by-twelve pattern holds for sympy (29 vs 31), matplotlib (24 vs 25), pytest (19 vs 20), astropy (13 vs 14), xarray (14 vs 15), sphinx (11 vs 12), scikit-learn (9 vs 10) and pylint (7 vs 8, whose Both 5 + Poly Only 1 = 6 also contradicts its own Polydev cell of 7). Since Eq. 3 defines Resolve@2 as exactly this union, Table 12 as printed contradicts the paper's central metric and must be regenerated from the released predictions.
12. **Table 10 and §5.6.** Table 10 lists nine repositories totalling $N = 431$ while the caption says only "Sorted by Instance Count"; requests ($N = 8$), seaborn ($N = 2$) and flask ($N = 1$) from Table 12 are silently dropped, and 58 further instances are unaccounted for even after adding them back. State the truncation, or print all twelve rows, and make the per-repository resolved counts reconcile with Table 4's 323 / 333 / 373.
13. **Table 6, all rows.** The averages do not follow from the totals at $N = 500$. Total Turns 44,048 over 500 instances is 88.1 per instance, not the stated 66.1 (the stated value implies $N \approx 666$); Polydev's 41,620 is 83.2, not 63.5 (implying $N \approx 655$). Likewise Average Duration 555.8 s over 500 instances is 77.2 hours, not the stated Total Duration of 102.8 hours (which implies 740.2 s average), and 819.9 s gives 113.9 hours against the stated 149.4 hours (implying 1,075.7 s). Both stated totals also contradict §4.5's "approximately 72 hours per full run" and the §5.7 Reproducibility Capsule's "72 hours runtime". Fix the denominators and reconcile all four statements.

14. **Table 7 against Table 6.** 655 consultations at an Average Consultation Duration of 293 seconds is 53.3 hours, but Table 6's incremental Polydev wall-clock over baseline is $149.4 - 102.8 = 46.6$ hours. Consultation alone cannot exceed the total slowdown it is said to cause. State whether consultations were blocking and whether the 293 s is wall-clock or summed across the two providers queried in parallel (§4.2).
15. **Table 11 and §6.2.** Every cell is a bare percentage with no denominator: "Multi-file changes required 78.2% / 21.8%", "Ambiguous problem statement 84.7% / 15.3%", "Simple pattern fix 42.1% / 57.9%". Add N per row, state whether the eight rows partition the 655 consultations of Table 7 or overlap, and extend the Appendix E labeling protocol (which covers only Table 7) to cover Table 11 and the §6.1 category counts (baseline-only $14+12+8+6 = 40$ and Polydev-only $18+12+11+9 = 50$, which do at least sum correctly). Also replace the Table 11 caption's formatting note "(Green: Italic; Red: Underline)" with an actual legend.
16. **§6.4, "Consultation Model Ablation", against Appendix C Table 13.** Normalise to a common N. As printed, +5 and +2 are on $N = 100$ while +10 is on $N = 500$, i.e. +5.0, +2.0 and +2.0 per hundred instances, which reverses the stated "synergistic effect" and the claim that both models give "nearly double the benefit of either alone". Report counts with denominators and CIs, and note that +10/500 is the same +2.0 pp that McNemar finds non-significant.
17. **§6.4, thinking-budget and turn-cap ablations.** The reported 58.2%, 61.8%, 54.2%, 60.4% and 63.2% are unattainable on the $N = 100$ subsets that Table 13 assigns to them. Correct N or the percentages, report counts, and state that the 64.6% endpoint comes from a different ($N = 500$) sample so the curves mix sample sizes.
18. **§4.6 and Table 3.** Replace "Commit Hash: [release-v1.0] (tagged at submission)" with an actual hash; give dated snapshot identifiers for gpt-5.2-codex and gemini-3-flash-preview rather than floating aliases; pin the SWE-bench Verified dataset revision; version the agent scaffold of §4.3 and the Polydev MCP server; and state the number of attempts per instance per policy (evidently 1) and the count of instances that hit the 250-turn or 128K-thinking caps.
19. **§7.2, recommendations 1-2, and §7.3.** The paper proposes confidence-based routing and a cascade as the deployable descendants of Resolve@2, with an "estimated 40-60% cost savings" that has no supporting measurement. The cascade in particular can be evaluated offline from the released predictions at zero additional inference cost: run baseline, and invoke Polydev only where the baseline patch fails its own reproduction tests. Reporting that number would convert the paper's ceiling into a floor, and would be its single most valuable addition.

20. **§3.5, heading and cross-reference.** §7.3 (limitation 3) points to "the theoretical analysis in Section 3.4", but the Pass@2 analysis is in §3.5; §3.4 is "When Does Model Consultation Help?". Fix the reference. While there, note in §3.5 that Pass@2 under the same harness-based selection is the metric-matched comparator for Resolve@2 — both consume two runs — and that the paper's own i.i.d. ceiling of 87.5% exceeds the reported 74.6%, so the unrun ablation is not a peripheral gap but the one that decides the thesis.
21. **§1.3 and Declarations.** Given that V.S.G. is affiliated with Polydev AI, the developer of the evaluated system, soften "To our knowledge, this is the first work to" and add a sentence in §4.2 noting that the consultation trigger heuristics of Listing 1 were authored by a party with an interest in the outcome. Declaring the interest, as the authors have done, is the right first step; insulating the design choices from it is the second.

5. Recommendation

Major revisions. The underlying experiment is worth reporting: 1,000 full agent runs over SWE-bench Verified with released trajectories and token logs is a real contribution, the complementarity structure of Table 5 (283 / 40 / 50) is a genuine and interesting finding, and the authors have been unusually forthright in §3.5, §7.3, and §8 about what they did not measure. But the paper as written does not support its title or its abstract. "Matching" rests on a label-aware union of two runs (Eq. 3, Table 4 row 3) compared against a quoted single-run leaderboard entry (Table 1 rank 1) whose CI it almost entirely overlaps; the mechanism the title advertises contributes +2.0 pp that the paper's own McNemar test cannot distinguish from zero; the cost advantage is computed with caching credited to one column and not the other; and Tables 6, 10, 12, and 13 do not reconcile with Table 4 or with each other. None of these are fatal to the work — the released artifacts should let the authors regenerate the tables and, more importantly, compute the cascade result that would give the paper a deployable headline number. I would be glad to review a revision.

Declarations

Potential competing interests: No potential competing interests to declare.